

Te lo resumo así parcial

Hoy, en **Te lo Resumo Así Nomás**: “*Parcial de Objetos*”. Esta es la historia de un un grupito de pibes del sábado a la tarde que se juegan la nota desarrollando un montón de líneas de código que no le van a servir a nadie nunca y encima tienen que hacerlo en tiempo record para ir a ver el superclásico (si es que la lluvia los deja).



Como parece que en la temporada #1 nos fue lo suficientemente bien, vamos a apostar a generar más resúmenes y ver si les gustan a nuestros espectadores, y nos metemos de lleno...

Resúmenes

Un resumen está conformado por una introducción y un conjunto de secciones, cada una de ellas tiene una duración y un speech, además algunas secciones son musicalizadas y tienen una canción asociada. Otras por ejemplo son referencias a otro resumen por lo tanto su speech y duración depende de la original, siendo el speech final el de esa sección más la que referencia, lo mismo aplica para la duración.

- 1) Queremos poder agregar una sección a un resumen. Además, nos gustaría que nos muestre cómo se haría esto, tanto la creación de la sección a agregar, como agregarla al resumen.
- 2) Queremos saber la duración de un resumen, que es la sumatoria de duraciones de la sección más el tiempo que tarda en decir la introducción (cantidad de palabras por 10).
- 3) Queremos saber si un resumen tiene cierta canción (lo único importante de la canción es el nombre)
- 4) Queremos saber si un resumen tiene una referencia a otro resumen
- 5) Queremos saber el libreto de un resumen, esto sería la introducción y todos los speech de las secciones concatenados por un "." (revisar [join](#))

Espectador y resúmenes

Queremos saber si un resumen le va a gustar a un espectador, esto ocurre cuando el resumen no dura más de lo que el espectador está dispuesto a ver y además cumple con ciertos requisitos dependiendo de qué tan exigente sea el espectador.

A los '*nuevos espectadores*', generalmente le gustan todos los resúmenes. A los '*followers*' que son los que vieron varios resúmenes, mientras tengan una canción que les guste les alcanza. Y a los '*fans*' (son los mas exigentes) solo le gustan los resúmenes que tienen todas las canciones que le gustan y tienen al menos una referencia a otro resumen.

Tener en cuenta que si venimos haciendo las cosas más o menos bien en nuestro canal, los nuevos espectadores eventualmente dejarán de serlo y se convertirían en *followers* o *fans*

- 1) Implemente lo que nos pidieron
- 2) Queremos saber si a un espectador le gustaría un resumen.
 - a) ¿Cómo creamos al espectador?
 - b) ¿Qué mensajes mandamos para que nos diga si le gustaría un resumen?

ResumenAsiNomas

Los resúmenes así nomás son muy similares al resto de los resúmenes, tienen una mini introducción del resumen y varios resúmenes de películas que tienen su propia introducción, secciones, etc. La duración del resumen así nomás es la misma que en los resúmenes comunes pero se le aplica un factor de velocidad definido en cada uno de estos. Tener en cuenta que este resumen tiene que comportarse igual al resto.

- 1) Agregue lo necesario para soportar los resúmenes así nomás.
- 2) ¿Hubo algún concepto que nos ayudó a agregar esto fácilmente? ¿Cómo te diste cuenta? Explique como funciona
- 3) Mostrar un ejemplo de instanciación de un resumenAsiNomas.

Suscripciones y popularidad

Tenemos que modelar las suscripciones del canal, para eso otro equipo de desarrollo nos dio el siguiente objeto que podemos utilizar, en el cual se van a ir agregando los suscriptores de nuestro canal:

```
object canalYoutube
  const suscriptores = #{}
  const resúmenes = #{}
  method agregarSuscriptor(suscriptor) { suscriptores.add(suscriptor) }
  method agregarResumen(resumen) { resúmenes.add(resumen) }
```

Nos interesa:

- 1) Agregar un resumen a nuestro canal. Es importante que al agregarlo, además se le agregue como pendiente de visualización a los suscriptores que podría gustarle ese resumen. Si no existiera ningún suscriptor al que le guste el resumen, no debemos agregarlo, ya que sería un resumen muy malo como para subirlo. Es importante que nos enteremos de esto.
- 2) Saber aquellos suscriptores adictos, es decir, los que vieron más de un 80% de los resúmenes.
- 3) Saber las canciones que piden los suscriptores (las canciones que les gusta). Es importante que sea sin repetidos.
- 4) Que los suscriptores una vez visto el resumen puedan darle, like o dislike. Nos gustaría saber el resumen con mas likes y el resumen con mas dislikes.

Nota:

Además de realizar el código que implemente lo pedido respetando los conceptos del paradigma de objetos, es necesario hacer el diagrama de clases.